120 AF/2172

PATENT
Customer No. 22,852
Attorney Docket No. 06502.0054-01

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:

Ann M. WOLLRATH et al.

Application No.: 10/015,801) Group Art Unit: 2172

Filed: December 17, 2001) Examiner: J. Fleurantin

For: METHOD AND APPARATUS FOR TRANSPORTING BEHAVIOR IN AN EVENT-BASED DISTRIBUTED SYSTEM

Mail Stop Appeal Brief--Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

TRANSMITTAL OF APPEAL BRIEF (37 C.F.R. 1.192)

Transmitted herewith in triplicate is the APPEAL BRIEF in this application with respect to the Notice of Appeal filed on February 17, 2004 and the Notification of Non-Compliance with 37 C.F.R. 1.192(c) mailed June 30, 2004.

This application is on behalf of

□ Small Entity ☒ Large Entity

Pursuant to 37 C.F.R. 1.17(c), the fee for filing the Appeal Brief is:

□ $165.00 (Small Entity)

☒ $330.00 (Large Entity)

TOTAL FEE DUE:

Notice of Appeal Fee $330.00

Extension Fee (if any)	$[Fee]
Total Fee Due	$330.00

☒ A check for $330.00 to cover the above fees was previously submitted on April 13, 2004.

PETITION FOR EXTENSION. If any extension of time is necessary for the filing of this Appeal Brief, and such extension has not otherwise been requested, such an extension is hereby requested, and the Commissioner is authorized to charge necessary fees for such an extension to our Deposit Account No. 06-0916. A duplicate copy of this paper is enclosed for use in charging the deposit account.

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: July 16, 2004

By: E. Niemeyer
Elizabeth A. Niemeyer
Reg. No. 52,070

PATENT
Customer No. 22,852
Attorney Docket No. 06502.0054-01

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:)
)
Ann M. WOLLRATH et al.)
)
Application No.: 10/015,801) Group Art Unit: 2172
)
Filed: December 17, 2001) Examiner: J. Fleurantin
)
For: METHOD AND APPARATUS FOR)
TRANSPORTING BEHAVIOR IN AN)
EVENT-BASED DISTRIBUTED)
SYSTEM)

Mail Stop Appeal Brief—Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

APPEAL BRIEF UNDER 37 C.F.R. § 1.192

In response to the June 30, 2004 Notification of Non-Compliance with 37 C.F.R. 1.192 (c) and in support of its Notice of Appeal filed February 17, 2004, and under 37 C.F.R. § 1.192, Appellants present in triplicate their Appeal Brief, wherein the fee of $330.00 under 37 C.F.R. § 1.17(c) was previously submitted on April 13, 2004. This is an appeal to the Board of Patent Appeals and Interferences from a decision finally rejecting claims 5–9 and 14–22. The appealed claims are set forth in the Appendix. If additional fees are required, please charge the deficiencies to Deposit Account No. 06-0916. If a fee is required for an extension of time under 37 C.F.R. § 1.136 and such fee is not accounted for above, Appellants petition for such an extension and request that the fee be charged to Deposit Account No. 06-0916.

I. **Real Party in Interest**

The real party in interest is Sun Microsystems, Inc., a Delaware corporation.

II. **Related Appeals and Interferences**

There are no known related pending appeals or interferences directly affected by or having a bearing on the decision in the pending appeal.

III. **Status of Claims**

Claims 5–9 and 14–22 have been finally rejected and are the subject of this appeal. Claims 1–4 and 10–13 have been previously cancelled. The claims on appeal are set forth under the heading APPENDIX. In the November 19, 2003 Final Office Action, the Examiner rejected claim 19 under 35 U.S.C. § 102(e) as anticipated by U.S. Patent 6,343,308 ("*Marchesseault*") and rejected claims 5–9, 14–18, and 20–22 under 35 U.S.C. § 103(a) as unpatentable over *Marchesseault* in view of U.S. Patent 5,101,346 ("*Ohtsuki*").

IV. **Status of Amendments**

Appellants filed a Preliminary Amendment on December 17, 2001, canceling claims 1–4 and 10–13, amending claim 5, and adding new claims 14 and 15. Appellants further filed a Supplemental Preliminary Amendment on October 18, 2002, clarifying the December 17, 2001 amendment to claim 5. Additionally, Appellants filed an Amendment on February 10, 2003, amending claim 5 and adding new claims 16–22.

V. Summary of Invention

Many distributed computer systems are organized to handle the occurrence of an event using an event notification model. (*See*, *e.g.*, Specification at p. 5, ll. 16–17.) Under the event notification model, a listener process registers interest in an event with a notifier process, which monitors occurrence of the event, and optionally designates a receiving process to be notified upon occurrence of the event. (*See*, *e.g.*, *id.* at p. 5, ll. 17–19.) Upon occurrence of the event, the notifier process notifies the receiving process, which may execute a function present in the address space (or physical machine) of the receiving process. (*See*, *e.g.*, *id.* at p. 5, l. 19–p. 6, l. 1.) In this model, control is determined by notifications sent in response to the occurrence of the designated events. (*See*, *e.g.*, *id.* at p. 6, ll. 1–3.)

In a distributed system, associating an event with a function to be run in response to the notification may be complicated. (*See*, *e.g.*, *id.* at p. 6, ll. 6–7.) The listener may be in one address space, the event may be generated in a second address space, and the notification may be sent to a third address space, making it difficult for the listened to ensure the receiving process will be able to respond to the event. (*See*, *e.g.*, *id.* at p. 6, ll. 7–13.)

One implementation consistent with the present invention addresses this problem by providing a system that allows transmission of registration of interest in an event in such a way as to allow subsequent notification of the event to include an object having methods that are invoked upon receipt of the notification. (*See*, *e.g.*, *id.* at p. 7, ll. 9–16; p. 9, ll. 7–11; p. 14, ll. 6–30.) In one aspect of the invention, when the notification is received, the methods may be executed as specified by the registration. (*See, e.g.*, *id.* at p. 8, ll. 5–10; p. 14, ll. 11–14.) Another implementation provides a message, which includes a registration object having closure, used to register an interest to notify an entity of an event and providing notification upon occurrence of the event. (*See, e.g.*, *id.* at p. 16, ll. 14–19.)

VI. Issues

The issues in this Appeal are:

(1) whether the Examiner's rejection of claim 19 under 35 U.S.C. § 102(e) as anticipated by *Marchesseault* can be affirmed when the reference does not teach every recitation of this claim, including receiving a registration of interest in an event, the registration including computer code, and transmitting a message including the computer code in response to the event as recited in claim 19; and

(2) whether the Examiner's rejection of claims 5–9, 14–18, and 20–22 under 35 U.S.C. § 103(a) as unpatentable over *Marchesseault* in view of *Ohtsuki* can be affirmed when the references do not support the Examiner's contention that these claims are obvious and there is no motivation to combine the references.

VII. Grouping of Claims

In the claims on appeal, claims 5, 18, 19, 20, 21, and 22 are the independent claims. The claims on appeal do not stand or fall together. These claims should be considered in three groups:

Group I:	19
Group II:	5–9, 14–17, 18, 20, and 22

Group III: 21

The claims have been placed in these groups due to their common subject matter. Appellants, however, have addressed the outstanding rejections in accordance with the rejections themselves instead of the above identified groupings.

VIII. **Argument**

Claim 5 recites a distributed computer system comprising a first virtual machine, a second virtual machine executing a process that receives, from the first virtual machine, a registration of interest in an event and transmits a message in response to the event, the registration of interest and the message including computer code, and a third virtual machine for receiving the message and executing the computer code. Claims 6–9 and 14–17 each depend from claim 5 and recite, respectively, a generic notify method message, a registration of interest including an identification of an event and an identification of a third virtual machine; separate virtual machines; implementing the computer code in an object; an event including a change in system state; a timer event, a mouse click event, and a disk access event; a virtual machine contained in a separate memory; and virtual machines run by separate processors.

Claim 18 recites a distributed computer system comprising a first virtual machine for transmitting a registration of interest in an event, the registration including computer code, a second virtual machine for transmitting a message including the computer code in response to the event, and a third virtual machine for executing the computer code transmitted in the message.

Claim 19 recites a method for executing computer code in a distributed computer system comprising receiving a registration of interest in an event, the registration including computer code, transmitting a message including the computer code in response to the event, and executing the computer code transmitted in the message.

Claim 20 recites a method for executing computer code in a distributed computer system comprising receiving from a first virtual machine a registration of interest in an event, the registration including computer code, transmitting from a second virtual machine a message including the computer code in response to the event, and executing on a third virtual machine the computer code transmitted in the message.

Claim 21 recites a method for sending event notifications in a system comprised of at least two entities, the method comprising providing, by a first one of the entities to a second one of the entities, a message including a registration object characterized as an object having closure, the second entity using the message to register an interest to notify an entity upon occurrence of an event, monitoring by the second entity for occurrence of the event, and providing by the second entity to another entity a notification upon occurrence of the event, whereby the first entity is not aware of functions available to the entity provided with the notification due to the closure of the registration object.

Claim 22 recites a computer object configured to operate on a machine, whereby the computer object comprises a method, and parameter data corresponding to the method, wherein the method and parameter data are adapted to be passed to a machine as part of an event registration message, the event registration message further includes event information identifying the event of interest and software information identifying a software entity to be notified upon occurrence of the event, and upon occurrence of the event, the method and parameter data execute to pass at least one of the computer object and reference to the computer object to the software entity.

A. The Rejection of Claim 19

Appellants appeal the rejection of claim 19 under 35 U.S.C. § 102(e) because *Marchesseault* does not support the Examiner's position set forth in the Office Action. For example, the Examiner cited entirely irrelevant portions of *Marchesseault* to support the rejection. Also, during an in-person interview with the Examiner, Appellants' representatives learned that the Examiner interprets the pending claim language "receiving registration of interest in an event," to mean nothing more than "computer code." The claim element recites, however, "receiving registration of interest in an event, the registration including computer code." Thus, under the Examiner's interpretation, the claim element would read "computer code, including computer code." Although the Examiner is entitled to interpret claim terms broadly, he is not entitled to interpret claim terms to the point of having no meaning at all. Accordingly, because *Marchesseault* does not support the rejection of claim 19 under 35 U.S.C. § 102(e), Appellants respectfully request that the Board reverse the Examiner's rejection of claim 19.

B. The Rejection of Claims 5–9, 14–17, 18, and 20–22

Appellants appeal the rejection of claims 5–9, 14–17, 18, and 20–22 under 35 U.S.C. § 103(a) as being unpatentable over *Marchesseault* in view of *Ohtsuki* because a *prima facie* case of obviousness has not been made by the Examiner. In particular, the references do not support the Examiner's position that the cited art suggests the Appellants' claimed invention. Further, there is no motivation to combine *Marchesseault*, which teaches loading a class interface to prevent calls to versions of object classes not recognized by a particular runtime system, with *Ohtsuki*, which teaches registering relations between the instruction processors and virtual machines to limit the range in virtual machines affected by operations of a multi-processor virtual machine.

C. The Rejection of Claim 19 Must Be Reversed Because *Marchesseault* Does Not Support the Examiner's Contention That the Reference Teaches All of the Recitations of the Claim.

In rejecting claim 19 under 35 U.S.C. § 102(e), the Examiner asserts that *Marchesseault* teaches every recitation of this claim. Appellants respectfully disagree. To reject a claim as anticipated, each claim element must be disclosed in a single reference and be arranged as required by the claim. (MPEP § 2131 (8th ed. Rev. 2003).)

Marchesseault discloses a system that allows Java applets written in any version of the Java programming language ("Java version") to be executed within any version of a Java Virtual Machine ("JVM"). (*Marchesseault*, col. 1, ll. 61–67.) In operation, the system downloads an applet written in a first Java version. (*Id.*, col. 2, ll. 23–27.) The applet includes a call to a second Java version object class. (*Id.*) When a client requests the applet, the server downloads the applet with a class interface to the client. (*Id.*, col. 2, ll. 29–33.) The applet is executed in a JVM located on the client and calls the second Java version object class. (*Id.*, col. 2, ll. 33–35.) The class interface verifies that each downloaded second Java version object class is identified by the loader within the JVM. (*Id.*, col. 2, ll. 36–44.) *Marchesseault* may automatically activate the applet based on one or more events (*e.g.*, activating a link, downloading an HTML page, etc.). (*Id.*, col. 4, ll. 16–19.)

The Examiner asserts that the disclosure in *Marchesseault's* claim 24 "is equivalent to for executing computer code in a distributed computer system as claimed, comprises receiving a registration of interest in an event, the registration including computer code." (May 22, 2003 Office Action at p. 3; November 19, 2003 Office Action at p. 3.)[1] Appellants respectfully disagree with the Examiner's position.

Moreover, "equivalence" is not a test for anticipation. To properly anticipate a claim, each claim element must be *disclosed* in a single reference. (MPEP § 2131.) Anticipation does not require identity of terminology, but it does require identity of invention. *Trintec Indus., Inc. v. Top-U.S.A. Corp.*, 295 F.3d 1292, 1294, 63 USPQ2d 1597, 1599 (Fed. Cir. 2002) ("novelty's identity requirement applies to claims"). A mere allegation that each element of claim 19 is merely met by "equivalents," is clearly not a *prima facie* case for anticipation. Appellants, nevertheless, address the Examiner's rejection.

The first version Java applets downloaded by *Marchesseault* generate calls to an object class associated with a different Java version. The applets are embedded in HTML documents using applet tags. The applets are activated "automatically upon the

[1] The body of *Marchesseault's* claim 24 recites in full, "a computer usable medium having first computer readable program code means embodied in said medium for downloading the applet to the client, wherein the applet includes each object class called within the applet; the computer usable medium having second computer readable program code means embodied in said medium for downloading to the client a class interface having at least one respective identifier for the object class associated with the second Java version of the Java Virtual Machine called by the applet; and the computer usable medium having computer readable program code means embodied in said medium, responsive to said first and second computer readable program code means, for executing the downloaded applet within the first Java version of the Java Virtual Machine without causing an error condition by the at least one call to an object class associated with the second Java version of the Java Virtual Machine."

loading of an HTML page by the web browser, by activating a link, or by user actions. Once activated, the applet begins to automatically download each class called within the applet." (*Marchesseault*, col. 4, ll. 8–22.)

The, "computer readable program code means that execute a downloaded applet within a first Java version," as disclosed in *Marchesseault*, is not the same as "receiving a registration of interest in an event, the registration including computer code," asserted by the Examiner. There is no disclosure in *Marchesseault* that a downloaded applet is either a registration of interest in an event or corresponds to receipt of registration of interest in an event. In fact, nowhere does *Marchesseault* disclose "receiving a registration of interest in an event," much less "the registration including computer code." In contrast to the Examiner's assertions, the applets disclosed by *Marchesseault* are neither a "registration of interest in an event" nor "a registration including computer code." Moreover, *Marchesseault's* applets cannot be the same as the "computer code" asserted by the Examiner because these applets are not transmitted in a message in response to an event. As admitted by the Examiner, *Marchesseault* teaches sending a request to activate the applet. (*See* May 22, 2003 Office Action, at p. 3, ll. 17–19.) Thus, an event merely initiates the request to activate the applet and does not include the applet itself.

Also, the Examiner's interpretation of *Marchesseault* ignores the recitations of claim 19. During an in-person interview conducted on January 28, 2004, Appellants' representatives specifically asked the Examiner to explain the rationale behind the rejection of claim 19, and the Examiner's citation to *Marchesseault* column 14, lines 23–32, as teaching "receiving a registration of interest in an event, the registration including computer code." The Examiner responded that he is entitled to read claims broadly, without offering further support for the rejection. Appellants' representatives further requested an elaboration on the Examiner's interpretation of "registration of interest in an event," as recited in claim 19. The Examiner indicated that his interpretation of registration of interest in an event is nothing more than "computer code." Accordingly, under the Examiner's interpretation, claim 19 recites, "computer code, including computer code." Not only is this nonsensical, but it also ignores the claim language.[2]

Therefore, *Marchesseault* does not support the rejection of claim 19 under 35 U.S.C. § 102(e). Further, the Examiner's interpretation of these recitations fails to give any meaning to the claim. Additionally, because *Marchesseault* fails to teach "a registration of interest in an event, the registration including computer code," it must also fail to teach transmitting a message including the computer code *in response to the event*, and "executing the computer code *transmitted in the message*," (emphasis added).

For at least these reasons, Appellants respectfully request that the Board reverse the rejection of claim 19 under 35 U.S.C. § 102(e).

[2] Appellants' representatives requested that the Examiner indicate he was interpreting "registration of interest in an event" to mean "computer code" on the Interview Summary, but the Examiner refused. Appellants' representatives also requested that the Examiner identify the specific issues discussed during the interview—*i.e.*, "receiving a registration of interest in an event, the registration including computer code" recited in claim 19—but the Examiner again refused. (January 28, 2004 Interview Summary.)

D. The Rejection of Claims 5-9, 14-18, and 20-22 Should Be Reversed Because *Marchesseault* and *Ohtsuki* Do Not Support the Examiner's Position That These Claims Are Obvious and There Is No Motivation to Combine the References.

Appellants traverse the rejection of claims 5–9, 14–17, 18, and 20–22 under 35 U.S.C. § 103(a) because a *prima facie* case of obviousness has not been made by the Examiner. To establish a *prima facie* case of obviousness under §103(a), each of three requirements must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to combine references or modify a reference. (MPEP § 2143 (8th ed. Rev. Feb. 2003).) Second, there must be a reasonable expectation of success. (*Id.*) Moreover, both of these requirements must "be found in the prior art, and not based on applicant's disclosure." (*Id.*) Third, the reference or references, taken alone or in combination, must disclose or suggest every element recited in the claims. (MPEP §2143.03.)

1. Group II: Claims 5–9, 14–17, 18, 20, and 22

a. Independent claim 5 and dependent claims 6–9 and 14–17

In the response filed August 20, 2003, Appellants demonstrated that *Marchesseault* in combination with *Ohtsuki* does not support the rejection of claim 5 and explained why the Examiner failed to establish a prima facie case of obviousness. (*See* August 20, 2003 Request for Reconsideration at 4–8.)

In response, the Examiner maintains the position that *Marchesseault* teaches all of the recitations of claim 5 except for "a second virtual machine executing a process that retrieves; and a third virtual machine." (November 19, 2003 Office Action, at p. 5, ll. 3–4.) The Examiner alleges,

> Marchesseault teaches . . . from the first machine, a registration of interest in an event and transmits a message in response to the event, the registration of interest and the message including computer code (thus, a computer usable medium having first computer readable program code means embodied in said medium for downloading the applet to the client wherein the applet includes each object class called within the applet, the computer usable medium having second computer readable program code means embodied in said medium for downloading the client a class interface having at least one respective identifier for the object class associated with the second Java version of the Java Virtual Machine called by the applet; which is readable as a registration of interest in an event, the registration of interest and the message including computer code)(see col. 14, lines 23-32).

(November 19, 2003 Office Action, at p. 4.) As explained above in connection with claim 19, *Marchesseault* merely discloses a system for downloading an applet, which may be automatically activated upon the loading of an HTML page by a web browser, the activation of a hyper-link, or other user actions. The downloading and activation of applets are not the same as or "readable as" the registration of interest in an event or the registration of interest and the message including computer code, as asserted by the Examiner. To interpret *Marchesseault's* disclosure of downloading and activation of applets either as a registration of interest in an event or as a registration of interest and the message including computer code, ignores the claim language. There is no disclosure of registration of interest in an event, and there is no disclosure of a registration of interest and the message including computer code.

The Examiner relies on *Ohtsuki* to compensate for the deficiencies of *Marchesseault* and asserts that it would have been obvious to combine these references to "provide a plurality of operating systems to run on an information processing system." (*Id.*, at p. 5, ll. 8–10.) Appellants disagree that these references teach or suggest the elements or processes as asserted by the Examiner and that there is any motivation to combine the references.

To begin with, *Ohtsuki* fails to make up for the deficiencies of *Marchesseault*. *Ohtsuki* discloses a virtual machine system including a main storage, which is allocated to a plurality of virtual machines. (*Ohtsuki*, col. 2, ll. 10–26.) The system also includes a system controller, which includes a registration table. (*Id.*, col. 2, ll. 27–29.) The registration table registers a correspondence between instruction processors and virtual machines such that operation of one real instruction processor only influences another real instruction processor assigned to the same virtual machine. (*Id.*, col. 2, ll. 30–45.) This avoids invalidation of a buffer storage and instruction processors. (*Id.*, col. 4, ll. 26–57.) Although *Ohtsuki* discloses multiple virtual machines, the reference does not teach or suggest the machine executing a process that receives a registration of interest in an event and transmits a message a message in response to the event, the registration of interest and the message including computer code, as asserted by the Examiner. Instead, *Ohtsuki* merely describes processes that inhibit buffer storage in instruction processes invalidations, none of which is associated with a registration of interest in an event or the message.

Also, there is no motivation to combine *Marchesseault* with *Ohtsuki*. According to the Examiner, "[s]uch modification would allow the teachings of Marchesseault and Ohtsuki to provide a plurality of operating systems to run on an information processing system, (see col. 1, lines 11-13)." (November 19, 2003 Office Action, at p. 5.) Appellants disagree. This is merely *Ohtsuki's* stated advantage. (*Ohtsuki*, col. 1, ll. 9–13.) *Ohtsuki* discloses a virtual machine system that includes a plurality of instruction processors and a main storage allocated to several virtual machines to minimize invalidation entries that occur when multiple virtual machines share a main storage. (*Ohtsuki*, col. 1, ll. 26–35 and col. 2, ll. 10–25.) As explained, *Marchesseault* is directed toward a multi-version applet/JVM system. Although *Ohtsuki* claims its invention provides the advantage of providing a plurality of operating systems to run on an information processing system, neither *Ohtsuki* nor *Marchesseault* provide motivation to implement a system for allowing multiple Java versions to run on a single virtual machine (as disclosed in *Marchesseault*) in a system including multiple virtual machines (as disclosed in *Ohtsuki*). Although *Ohtsuki* may allege an advantage from its invention, the cited art does not suggest either any advantage resulting from the *combination* of, or any reason to combine, *Marchesseault* with *Ohtsuki*.

Because the proposed combination of *Ohtsuki* and *Marchesseault* fails to support the rejection of claim 5 under 35 U.S.C. § 103(a), a *prima facie* case of obviousness has not been established by the Examiner. Accordingly, Appellants respectfully request that the Board reverse the rejection of the claim 5.

Claims 6–9 and 14–17 depend from claim 5. As explained, the rejection of claim 5 is unsupported by *Marchesseault* and *Ohtsuki*. Accordingly, the rejection of claims 6–9 and 14–17 are also unsupported by these references for at least the same reasons set forth in connection with claim 5. Therefore, Appellants respectfully request that the Board reverse the rejection of these claims.

b. Independent claim 18

Appellants submit that the Examiner failed to establish a *prima facie* case of obviousness for at least the reasons given above with respect to claim 5. For instance, *Marchesseault* and *Ohtsuki* alone and in combination fail to teach or suggest a first virtual machine for transmitting a registration of interest in an event, the registration including computer code, a second virtual machine for transmitting a message including the computer code in response to the event, and a third virtual machine for executing the computer code transmitted in the message, as asserted by the Examiner. Because the references fail to support the Examiner's assertions, there can be no reasonable expectation of success in achieving the combination asserted by the Examiner. Further, as explained above, neither reference provides motivation to combine the references. Accordingly, the Examiner failed to establish a *prima faci*e case of obviousness in rejecting claim 18 over these references. Therefore, Appellants respectfully request that the Board reverse the rejection of claim 18.

c. Independent claim 20

Appellants submit that the Examiner failed to establish a *prima facie* case of obviousness for at least the reasons given above with respect to claim 5. For instance, *Marchesseault* and *Ohtsuki* fail to disclose or suggest, receiving from a first virtual machine a registration of interest in an event, the registration including computer code, transmitting from a second virtual machine a message including the computer code in response to the event, and executing on a third virtual machine the computer code transmitted in the message, as asserted by the Examiner. Because the references fail to support the Examiner's assertions, there can be no reasonable expectation of success in achieving the combination asserted by the Examiner. Further, as previously explained, neither reference provides motivation to combine the references. Accordingly, the Examiner failed to establish a *prima faci*e case of obviousness in rejecting claim 20 over these references. Therefore, Appellants respectfully request that the Board reverse the rejection of claim 20.

d. Independent claim 22

The Examiner alleges:

> As per claim 22, in addition to the discussion in claim 5, Marchesseault further teaches the event registration message further includes event information identifying the event of interest and software information identifying a software entity to be notified upon occurrence of the event (thus, verifying that each identified object class associated with the second Java version of the Java Virtual Machine has a respective identifier in the class interface; which is readable as an event information identifying the event of interest and software information identifying a software entity to be notified upon occurrence of the event)(see col. 14, lines 4-7), and
>
> upon occurrence of the event, the method and parameter data execute to pass at least one of the computer object and reference to the computer object to the software entity (thus, methods, systems and computer program products for executing an applet within a first Java version of a Java Virtual Machine running on a client, wherein the applet includes at least one call to a second Java version object class, are provided, a request to activate an applet is made to a server hosting the applet from a client; which is readable as upon occurrence of the event, the method and parameter data execute to pass at least one of the computer object and reference to the computer object to the software entity)(see col. 2, lines 24-29).

(May 22, 2003 Office Action, at pp. 8–9). Appellants disagree with the Examiner's characterization of *Marchesseault*, which discloses a system for allowing multiple Java versions to run on a single virtual machine. The object class disclosed by *Marchesseault* includes an identifier that permits a class interface to determine a Java version of an applet. In particular, Appellants submit that the disclosure of *Marchesseault* is not "readable as an event information identifying the event of interest and software information identifying a software entity to be notified upon occurrence of the event," or "readable as upon occurrence of the event, the method and parameter data execute to pass at least one of the computer object and reference to the computer object to the software entity," as alleged by the Examiner. (November 19, 2003 Office Action, at p. 8.)

Moreover, *Ohtsuki* does not make up for the deficiencies of *Marchesseault*. The Examiner has merely cited irrelevant lengthy portions of *Marchesseault's* disclosure and alleged they are "readable" on recited claim elements. *Ohtsuki* merely describes processes that inhibit buffer storage in instruction processes invalidations. It does not disclose or suggest an event registration message that includes event information identifying the event of interest or software information identifying a software entity to be notified upon occurrence of the event, as asserted by the Examiner.

Because neither *Marchesseault* or *Ohtsuki* supports the Examiner's rejection, there can be no finding of obviousness of claim 22 in view of the cited references. Further, as explained with respect to claim 5, *Marchesseault* with *Ohtsuki* fail to provide a suggestion that motivates one skilled in the art to combine the references that result in Appellants' claimed invention. Accordingly, because the Examiner failed to establish a *prima facie* case of obviousness, Appellants respectfully request that the Board reverse the rejection of claim 22.

2. Group III: Independent claim 21

According to the Examiner:

> As per claim 21, in addition to the discussion in claim 5, Marchesseault further teaches a message including a registration object characterized as an object having closure (thus, wherein the application includes at least one call to an object class of a second version of the runtime system; which is equivalent to a message including a registration

> object characterized as an object having closure)(see [*Marchesseault*,] col. 2, lines 4-6).

(November 19, 2003 Office Action, at pp. 7–8). Appellants disagree with the Examiner's characterization of *Marchesseault*, which merely discloses a system for allowing multiple Java versions to run on a single virtual machine. Although the Examiner declares that "the application includes at least one call to an object class of a second version of the runtime system" is equivalent to "a message including a registration object characterized as an object having closure," the Examiner provides nothing to support this statement. There is no demonstration that an application including a call to an object class of a second version is readable on "a message including a registration object characterized as an object having closure," as asserted by the Examiner.

To the extent the Examiner relies on the description of *Marchesseault* and *Ohtsuki* given with respect to claim 5 to disclose a second entity using the message to register an interest to notify an entity upon occurrence of an event, monitoring by the second entity for occurrence of the event, and providing by the second entity to another entity a notification upon occurrence of the event, Appellants demonstrated above that neither reference supports the Examiner's position that they disclose or suggest registering an interest to notify an entity upon occurrence of an event. Accordingly, the Examiner's conclusion that these references disclose or suggest elements or processes that monitor for an occurrence of an event or providing a notification upon occurrence of the event is incorrect and unsupported by the cited art.

Because neither reference supports the rejection of claim 21 under 35 U.S.C. § 103(a), as asserted by the Examiner, and, as explained with respect to claim 5, *Marchesseault* with *Ohtsuki* fail to provide a suggestion that motivates one skilled in the art to combine the references as asserted by the Examiner, no *prima facie* case of obviousness has been established. Accordingly, Appellants respectfully request that the Board reverse the rejection of claim 21.

IX. Conclusion

For at least the foregoing reasons, the Board should reverse the rejection of claim 19 under 35 U.S.C. § 102(e) and of claims 5–9, 14–18, and 20–22 under 35 U.S.C. § 103(a).

To the extent any extension of time under 37 C.F.R. § 1.136 is required to obtain entry of this Appeal Brief, such extension is hereby respectfully requested. If there are any fees due under 37 C.F.R. §§ 1.16 or 1.17 which are not enclosed herewith, including any fees required for an extension of time under 37 C.F.R. § 1.136, please charge such fees to our Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: July 16, 2004

By: E. Niemeyer
Elizabeth A. Niemeyer
Reg. No. 52,070

Post Office Address
(to which correspondence is to be sent)

Finnegan, Henderson, Farabow,
Garrett & Dunner, L.L.P.
1300 I Street, N.W.
Washington, D.C. 20005
(202) 408-4000

Appendix

5. A distributed computer system comprising:

a first virtual machine;

a second virtual machine executing a process that receives, from the first virtual machine, a registration of interest in an event and transmits a message in response to the event, the registration of interest and the message including computer code; and

a third virtual machine for receiving the message and executing the computer code.

6. The distributed system of claim 5, wherein the message transmitted by the second virtual machine is a generic notify method.

7. The distributed system of claim 5, wherein the registration of interest by the second virtual machine includes an identification of the event and an identification of the third virtual machine.

8. The distributed system of claim 5, wherein each said virtual machine is stored on a separate computer system.

9. The distributed system of claim 5, wherein the computer code is implemented in an object.

14. The distributed computer system of claim 5, wherein the event includes a change in system state.

15. The distributed computer system of claim 5, wherein the event includes one selected from the group consisting of a timer event, a mouse click event, and a disk access event.

16. The distributed computer system of claim 5, wherein each said virtual machine is contained in a separate memory.

17. The distributed computer system of claim 5, wherein each said virtual machine is run by a separate processor.

18. A distributed computer system comprising:

a first virtual machine for transmitting a registration of interest in an event, the registration including computer code;

a second virtual machine for transmitting a message including the computer code in response to the event; and

a third virtual machine for executing the computer code transmitted in the message.

19. A method for executing computer code in a distributed computer system comprising:

receiving a registration of interest in an event, the registration including computer code;

transmitting a message including the computer code in response to the event; and

executing the computer code transmitted in the message.

20. A method for executing computer code in a distributed computer system comprising:

receiving from a first virtual machine a registration of interest in an event, the registration including computer code;

transmitting from a second virtual machine a message including the computer code in response to the event; and

executing on a third virtual machine the computer code transmitted in the message.

21. A method for sending event notifications in a system comprised of at least two entities, the method comprising:

providing, by a first one of the entities to a second one of the entities, a message including a registration object characterized as an object having closure;

the second entity using the message to register an interest to notify an entity upon occurrence of an event;

monitoring by the second entity for occurrence of the event; and

providing by the second entity to another entity a notification upon occurrence of the event,

whereby the first entity is not aware of functions available to the entity provided with the notification due to the closure of the registration object.

22. A computer object configured to operate on a machine, wherein the computer object comprises:

a method; and

parameter data corresponding to the method, wherein:

the method and parameter data are adapted to be passed to a machine as part of an event registration message,

the event registration message further includes event information identifying the event of interest and software information identifying a software entity to be notified upon occurrence of the event, and

upon occurrence of the event, the method and parameter data execute to pass at least one of the computer object and reference to the computer object to the software entity.